UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/741,502	12/19/2000	Theodore S. Hills	TAJ-0001	5730

23413 7590 03/26/2004

CANTOR COLBURN, LLP
55 GRIFFIN ROAD SOUTH
BLOOMFIELD, CT 06002

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2124	4

DATE MAILED: 03/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/741,502	HILLS, THEODORE S.	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 December 2000*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-48* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-48* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *19 December 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed December 19, 2000.

Claims 1-48 have been submitted for examination.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 13, 29 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 13 and 29 recite the limitation "and the instruction set" in line 3, line 4, respectively. There is insufficient antecedent basis for this limitation in the claim. The examiner will treat the limitation as if it were 'an instruction set'.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 13-14, and 29-30 are rejected under 35 U.S.C. 102(b) as being anticipated by Van Preat et al., USPN: 5,854,929 (hereinafter VanPreat).

As per claim 13, VanPreat discloses a method of compilation comprising

generating a description of computer architecture as a first library (e.g. *LIB format* - col. 22, lines 7-27; col. 18, lines 25-64), the description including software-visible objects (*vertices,*

storage elements, representing operations - col. 5, lines 5-24) and an instruction set (e.g. *nML description* – col. 18, line 35 to col. 19, line 32; col. 22, lines 28-30 – Note: description to specify processor instruction set reads on description including instruction set);

implementing a record of high level objects (e.g. Fig. 6; *cdfg files* , col. 22, lines 7-27) using the description stored in the first library; and binding a source program to implementations on the record of high-level objects to produce machine instructions dependent on the computer architecture (e.g. col. 22, line 61 to col. 23, line 21).

VanPreat does not explicitly disclose that the record of high-level objects being implemented with a source program is a second library; however, Vanpreat discloses generating of ISG specification in form of a library (e.g. *Library L* - col. 11, lines 36-55) to translate data of the first library (col. 22, lines 28-55); hence has disclosed such record to be a second library.

As per claim 14, VanPreat discloses a method of re-targeting comprising the steps of generating (first description/first library); generating a second description (second library) as mentioned in claim 13 from above; these steps being therefore rejected using the same corresponding rejections as set forth therein.

Further, VanPreat discloses implementing first description using the second library to produce a third library and binding a source program to implementations on the third library (e.g. *new lib and .isg files* - col. 22, lines 53-55).

As per claim 29, this is medium/machine-readable code version of claim 13; hence is rejected using the corresponding rejections as set forth therein.

As per claim 30, this is medium/machine-readable code version of claim 14; hence is rejected using the corresponding rejections as set forth therein.

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-7,10-12, 15, 17-23, 26, 28-31, and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Van Preat et al., USPN: 5,854,929 (hereinafter VanPreat), in view of Killian et al., USPN: 6,477,683 (hereinafter Killian), and further in view of Conner et al., USPN: 5,428,792 (hereinafter Conner)

As per claim 1, VanPreat discloses a computer programming method comprising:

describing data types (e.g. col. 5, lines 5-24 - Note: each operations represented as vertices is a type) as distinctly from the classes or interfaces (Note: declaration of data type is different from declaration of class or interfaces);

describing subtype (e.g. col. 11, lines 36-42) and supertype relationships (e.g. Fig. 15) between the types; and

describing representations of values (e.g. *vertices, storage elements, representing operations* - col. 5, lines 5-24; col. 11, lines 9-34) of the types (Note: each operations as vertices is associated type) as states of operations/objects defined from the model.

But VanPreat does not explicitly disclose that the data types are abstract without default or implicit implementations. However, VanPreat discloses abstract and neutral operation type (e.g. SUB – Fig. 5) from which another subtype can be generated, i.e. defining such abstract type

without default or implicit implementation. The abstracting of data type for object-oriented modeling and hierarchization of types in metadata language specification (e.g. UML, MOF, JCOM) was a well-known concept in the software development at the time the invention was made. Further, in a method to compile code for targeting a configurable processor circuitry analogous to VanPreat's generating of code for a programmable processor, Killian defines abstract data type with an object-oriented programming language (e.g. col. 20, lines 35-40). In case VanPreat does not already provide abstract data type to support the high-level model specification of the application, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide abstract data type as taught by Killian because abstracting of data type enable further generation of subtypes thereby enhancing the process of abstracting the specification of the application model such as evidenced by object-oriented data type level of abstraction suggested by Killian and so well-known in most object-oriented modeling methodologies.

Nor does VanPreat disclose that representing the values of the types as states of operations/objects being modeled is actually representing those types as states of classes of objects with interfaces. VanPreat however classifies targeted instruction architecture into classes (e.g. Fig. 11) using nML language, i.e. hierarchization of instruction set (e.g. col. 18, lines 58 to col. 19, line 59) and Killian, using HDL/Verilog language (Fig. 6; col. 16, lines 13-52) to classify instruction set for targeted processors, suggests C++ to implement the executable code (e.g. col. 40, lines 19-30 – Note: classes and interfaces are implicitly disclosed in object-oriented language). The object-oriented approach in the modeling and classification of instructions and use of object-oriented approach is hence suggested. Further, Conner, in a method analogous to

the nML by VanPreat or HDL by Killian to use a definition language (Conner: *OIDL* – col. 2, lines 2-24) for developing a neutral modeling (i.e. abstract data type or supertype) specification language so to retarget various object code application, discloses object-oriented model with classes and interfaces (e.g. Fig. 3-15). In case object-oriented programming such as taught by Conner were available to the modeling and compilation scheme as disclosed by VanPreat (combined with Killian), it would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the states of operations/objects being modeled as object-oriented classes of objects with interfaces as suggested by Killian and evidenced by Conner, because the use of object class implemented by interfaces would further enhance the code generation scheme by VanPreat with all the benefits so well-appreciated in a object-oriented language (e.g. inheritance, polymorphism, encapsulation, reusability, etc.).

As per claim 2, VanPreat (combined with Killian) discloses software-visible physical objects (e.g. col. 3, line 65 to col. 4, line 54; col. 5, lines 5-24) and an instruction set (e.g. col. 8, line 38 to col. 9, line 15; col. 11, line 1 to col. 12, line 60) using instructions constructs classes ; but does not disclose object-oriented classes implemented by a computer. But this limitation has been addressed in claim 1 above using both Killian and Conner's teachings of object-oriented constructs being implemented for computer code generation.

As per claim 3, VanPreat does not disclose multiple classes of pointer objects; but in view of Killian's or Conner's suggestion in claim 1 about using C++ to implement the retargetable code, the concept of using C++ pointer object is thereby disclosed.

As per claim 4, the tree-like organization of operations associated with class of operands in VanPreat (e.g. Fig. 3ac, 4, 5) combined with the teachings of Killian and Conner using object-oriented approach disclose non-sequential ways of calling routines and transferring control.

As per claim 5, VanPreat (combined with Killian/Conner) discloses describing at least one interfaces, classes, enumerations, subroutines as classes of objects (Note: these are implicitly disclosed in C++ or SmallTalk type of programming).

As per claims 6 and 7, VanPreat discloses invoking statements in a compilation

with results into a output of a compiler (col. 22, lines 61-64) and

to interpret literals in a input of a compiler (e.g. col. 19, lines 13-15; col. 22, lines 15-60; col. 23, lines 2-28).

As per claims 10 and 11, the limitation as to formal arguments in C++ or object-oriented language routines encompassing arguments such as type, classes, interfaces, data attributes are implicitly disclosed in Killian/Conner teachings or common object-oriented programming language specification.

As per claim 12, in view of the concept that a class is an argument provided in the list of formal parameters as mentioned in the rejection of claims 10-11, the teaching that a class is being parametized as part of such listing reads on subroutines being parametized classes.

As per claim 15, VanPreat discloses a computer programming method comprising describing software-visible physical objects (*vertices, storage elements, representing operations* - col. 5, lines 5-24) and an instruction set (e.g. *nML description* – col. 18, line 35 to col. 19, line 32; col. 22, lines 28-30); but does not disclose the limitations as to using object-oriented classes and identifying when classes are implemented by a computer.

These limitations have been addressed in claim 1 and 2 above using the combined teachings of Killian and Conner.

As per claim 17, this is storage medium and machine-readable code version of claim 1; hence is rejected using the corresponding rejections as set forth therein.

As per claims 18-23, these claims correspond to claims 2-7, respectively; hence are rejected with the corresponding rejections as set forth therein.

As per claims 26-28, these claims correspond to claims 10-12, respectively; hence are rejected with the corresponding rejections as set forth therein.

As per claim 31, this is storage medium and machine-readable code version of claim 15; hence is rejected using the corresponding rejections as set forth therein.

As per claim 47, this claim is propagation medium signal version of claim 15; hence is rejected using the corresponding rejections as set forth in claim 15; and further includes a propagation medium/encoded signal limitation. Official notice is taken that the use of propagation medium encoding a signal embodying a code with instructions was a well-known concept at the time the invention was made. Hence, for an ordinary skill in the art at the time of the invention was made, it would have been obvious to provide the method as disclosed by VanPreat with a propagation medium to support the signal embodying the code made for VanPreat's product because such medium can facilitate the distribution of software throughout the internet to distributors the most efficient way as has been appreciated in the realm of internet distributed products and/or online software sale/maintenance/advertisement/support.

8. Claims 33-39, and 42-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Van Preat et al., USPN: 5,854,929.

As per claim 33, this is propagation medium/encoded signal version of claim 1; hence is rejected using the corresponding rejections as set forth therein. VanPreat does not provide a signal propagated over a propagation medium. Official notice is taken that the use of propagation medium encoding a signal embodying a code with instructions was a well-known concept at the time the invention was made. Hence, for an ordinary skill in the art at the time of the invention was made, it would have been obvious to provide the method as disclosed by VanPreat with a propagation medium to support the signal embodying the code made for VanPreat's product because such medium can facilitate the distribution of software throughout the internet to distributors the most efficient way as has been appreciated in the realm of internet distributed products and/or online software sale/maintenance/advertisement/support.

As per claims 34-39, these claims correspond to claims 2-7, respectively; hence are rejected with the corresponding rejections as set forth therein.

As per claims 42-44, these claims correspond to claims 10-12, respectively; hence are rejected with the corresponding rejections as set forth therein.'

As per claim 45, this claim includes the propagation medium signal as addressed in claim 33; and is propagation medium signal version of claim 13; hence is rejected using the corresponding rejections as set forth in claim 13.

As per claim 46, this claim includes the propagation medium signal as addressed in claim 33; and is propagation medium signal version of claim 14; hence is rejected using the corresponding rejections as set forth in claim 14.

9. Claims 8-9, and 24-25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Van Preat et al., USPN: 5,854,929, Killian et al., USPN: 6,477,683, and Conner et al., USPN:

5,428,792; as applied to claim 1(for claims 8-9), claim 17 (for 24-25) and claim 33 (for 40-41); and further in view of Chen, USPN: 6,003,038 (hereinafter Chen).

As per claim 8, from the combination VanPreat/Killian/Conner, Killian and Conner discloses definition language to specify the application targeted for various execution environments using object-oriented; but neither VanPreat, Killian, or Conner disclose deriving variable classes from a constant class; and using a single descriptor to describe one of such variable classes and the constant class. However, VanPreat discloses supertype (Fig. 5), and Conner disclose a neutral metadata class (*default metaclass 650* – Fig. 6) and its interface (e.g. col. 25, lines 45-58); suggesting a type of unimplemented and undifferentiated class from which to derive other subclass. Official notice is taken that the identification of constant data or constant classes so to segregate them in a constant record in object-oriented programming language was a known concept at the time the invention was made. Analogous to VanPreat/Killian, Chen, in a method using a class structure to provide a instruction set for an object-oriented processor to be executed in that object-oriented operating environment, discloses a constant pool from which to derive instance of classes or data (e.g. col. 6, lines 9-35; Fig. 4). In view of the suggested use of object-oriented approach to derive objects or instance of subclasses from a more generic class as mentioned above, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide a constant record as taught by Chen or by official notice to VanPreat/Killian/Conner's method in case such method does not already include one such constant pool or class. One of ordinary skill would be motivated to do so because providing a constant class would alleviate extraneous proliferation of class instances which are unchanged; and thus by describing such constant class along with its derived children

(or variable classes) by a single descriptor, like a superclass pointer or metadata class pointer (or single descriptor) as suggested by VanPreat or Conner, would enhance even further the optimizing process as has been introduced by Chen.

As per claim 9, the object-oriented approach by Conner implicitly discloses class and interfaces being implemented; and along with the rationale as to why a common constant class be described to encompass both constant class as suggested by Chen and its derived children classes as set forth above, the limitation to provide a constant interface and representing a constant interface and its children variable interfaces by a single descriptor would also have been obvious for the same rationale and implicit teachings as mentioned above.

As per claims 24-25, these claims correspond to claims 8-9, respectively; hence are rejected with the corresponding rejections as set forth therein.

10. Claims 16, 32, 40-41 and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Van Preat et al., USPN: 5,854,929, and Conner et al., USPN: 5,428,792; and further in view of Chen, USPN: 6,003,038 (hereinafter Chen)

As per claim 16, VanPreat does not disclose variable classes derived from a constant class and describing variable classes and constant class with a single descriptor. However, VanPreat discloses supertype (Fig. 5), and Conner discloses object-oriented implementation (re claim 1) with establishing of a neutral metadata class (*default metaclass 650* – Fig. 6) and its interface (e.g. col. 25, lines 45-58); suggesting a type of unimplemented and undifferentiated class from which to derive other subclass; but VanPreat does not disclose the limitations as to using object-oriented classes and identifying when classes are implemented by a computer. In

view of Conner's teachings as mentioned in claim 1, this limitation to use object-oriented and class objects would have been obvious for the same rationale as set forth therein.

It is well recognized that the provision to identify of constant data or constant classes so to segregate them in a constant record in many object-oriented programming language (e.g. Java) was a well known concept at the time the invention was made. In view of Chen's teachings on constant pool as set forth in claims 8-9 from above, the limitation as to deriving variable classes from a constant class and describing variable classes and constant class with a single descriptor would have been obvious for the same reasons as set forth in claims 8-9 above.

As per claim 32, this is storage medium/machine-readable code version of claim 16; hence is rejected using the corresponding rejections as set forth therein.

As per claims 40-41, these claims include the propagation medium signal as addressed in claim 33; and further include limitations corresponding to claims 8-9, respectively, or claim 16; hence are rejected with the rationale of claim 16.

As per claim 48, this claim includes the propagation medium signal as addressed in claim 33; and is storage medium/machine-readable code version of claim 16; hence is rejected using the corresponding rejections as set forth therein.

Conclusion

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 872-9306 (for formal communications intended for entry)

or: (703) 746-8734 (for informal or draft communications, please consult Examiner before using this number)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
March 19, 2004

TODD INGBERG
PRIMARY EXAMINER